International Journal of Computer Vision (2023) 131:813–833
https://doi.org/10.1007/s11263-022-01713-6

Overcoming the Domain Gap in Neural Action Representations

Semih Günel[1,2] · Florian Aymanns[2] · Sina Honari[1] · Pavan Ramdya[2] · Pascal Fua[1]

Received: 29 April 2022 / Accepted: 7 November 2022 / Published online: 19 December 2022

Abstract
Relating behavior to brain activity in animals is a fundamental goal in neuroscience, with practical applications in building robust brain-machine interfaces. However, the domain gap between individuals is a major issue that prevents the training of general models that work on unlabeled subjects. Since 3D pose data can now be reliably extracted from multi-view video sequences without manual intervention, we propose to use it to guide the encoding of neural action representations together with a set of neural and behavioral augmentations exploiting the properties of microscopy imaging. To test our method, we collect a large dataset that features flies and their neural activity. To reduce the domain gap, during training, we mix features of neural and behavioral data across flies that seem to be performing similar actions. To show our method can generalize further neural modalities and other downstream tasks, we test our method on a human neural Electrocorticography dataset, and another RGB video data of human activities from different viewpoints. We believe our work will enable more robust neural decoding algorithms to be used in future brain-machine interfaces.

Keywords Animal pose estimation · Two-photon microscopy · Action recognition · Self-supervised learning

1 Introduction

Neural decoding of action, the accurate prediction of behavior from brain activity, is a fundamental challenge in neuroscience with important applications in the development of robust brain machine interfaces (Ahmed et al., 2021; Spampinato et al., 2017; Palazzo et al., 2018, 2021). Recent technological advances have enabled simultaneous recordings of neural activity and behavioral data in experimental animals and humans (Dombeck et al., 2007; Seelig et al., 2010; Chen et al., 2018; Pandarinath et al., 2018; Ecker et al., 2010; Topalovic et al., 2020; Urai et al., 2021). Nevertheless, our understanding of the complex relationship between behavior and neural activity remains limited.

A major reason is that it is difficult to obtain many recordings from mammals and a few subjects are typically not enough to perform meaningful analyses (Pei et al., 2021). This is less of a problem when studying the fly *Drosophila melanogaster*, for which long neural and behavioral datasets can be obtained for many individual animals (Fig. 1). Nevertheless, current supervised approaches for performing neural decoding (Nakagome et al., 2020; Glaser et al., 2020) still do not generalize well across subjects because each nervous system is unique (Fig. 2). This creates a significant domain-gap that necessitates tedious and difficult manual labeling of actions. Furthermore, a different model must be trained for each individual subject, requiring more annotation and overwhelming the resources of most laboratories.

Another problem is that experimental neural imaging data often has unique temporal and spatial properties. The slow decay time of fluorescence signals introduces temporal artifacts. Thus, neural imaging frames include information about an animal's previous behavioral state. This complicates decoding and requires specific handling that standard machine learning algorithms do not provide.

P. Ramdya, P. Fua: These authors contributed equally to this work.

✉ Semih Günel
semih.gunel@epfl.ch

Florian Aymanns
florian.aymanns@epfl.ch

Sina Honari
sina.honari@epfl.ch

Pavan Ramdya
pavan.ramdya@epfl.ch

Pascal Fua
pascal.fua@epfl.ch

[1] Computer Vision Lab, EPFL, Lausanne, Switzerland

[2] Neuroengineering Lab, EPFL, Lausanne, Switzerland

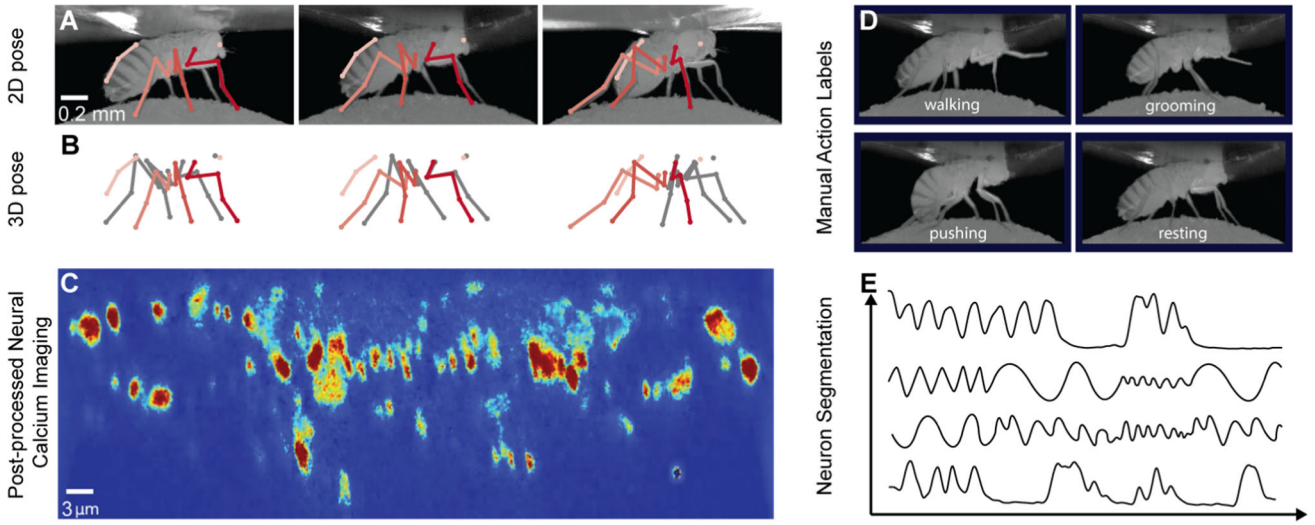

Fig. 1 Our Motion Capture and Two-Photon (MC2P) Dataset. A tethered fly (*Drosophila melanogaster*) is recorded using six multi-view infrared cameras and a two-photon microscope. The resulting dataset includes the following. **A** 2D poses extracted from different views (only three are shown), calculated on grayscale images. **B** 3D poses triangulated from the 2D views. **C** Synchronized, registered, and denoised single-channel fluorescence calcium imaging data using a two-photon microscope. Shown are color-coded activity patterns for populations of descending neurons from the brain. These carry action information (red is active, blue is inactive). **D** Annotations for eight subjects of eight different behaviors, four of which are shown here. **E** Manual neural segmentation has been performed to extract neural activity traces for each neuron. We will release our MC2P publicly. Examples videos of selected actions and multi-modal data are in the Supplementary Material

To address these challenges, we propose to learn neural action representations—embeddings of behavioral states within neural activity patterns—in an self-supervised fashion. To this end, we leverage the recent development of computer vision approaches for automated, markerless 3D pose estimation (Günel et al., 2019; Nath et al., 2019) to provide the required supervisory signals without human intervention. We first show that using contrastive learning to generate latent vectors by maximizing the mutual information of simultaneously recorded neural and behavioral data modalities is not sufficient to overcome the domain gap between subjects and to generalize to unlabeled subjects at test time (Fig. 3B). To address this problem, we introduce two sets of techniques:

1. To close the domain gap between subjects, we leverage 3D pose information. Specifically, we use pose data to find sequences of similar actions between a source and multiple target subjects. Given these sequences, we mix and replace neural or behavioral data of the source subject with the ones composed of multiple target subjects. To make this possible, we propose a new Mixup strategy which merges selected samples from multiple target animals, practically hiding the identity information. This allows us to train our decoder to ignore subject identity and close the domain gap.
2. To mitigate the slowly decaying calcium data impact from past actions on neural images, we add simulated randomized versions of this effect to our training neural images in the form of a temporally exponentially decaying random action. This trains our decoder to learn the necessary invariance and to ignore the real decay in neural calcium imaging data. Similarly, to make the neural encoders robust to imaging noise resulting from low image spatial resolution, we augment random sequences into sequences of neural data to replicate this noise.

The combination of these techniques allowed us to bridge the domain gap across subjects in an unsupervised manner (Fig. 3D), making it possible to perform action recognition on unlabeled subjects better than earlier techniques, including those requiring supervision (Glaser et al., 2020; Batty et al., 2019; Kostas et al., 2021). To test the generalization capacity of neural decoding algorithms, we record and use MC2P dataset, which we will make publicly available (Aymanns et al., 2022) [1]. It includes two-photon microscope recordings of multiple spontaneously behaving *Drosophila*, and associated behavioral data together with action labels.

Finally, to demonstrate that our technique generalizes beyond this one dataset, we tested it on two additional ones. One dataset features neural ECoG recordings and 2D pose

[1] The dataset can be accessed from https://github.com/semihgunel/mc2p.

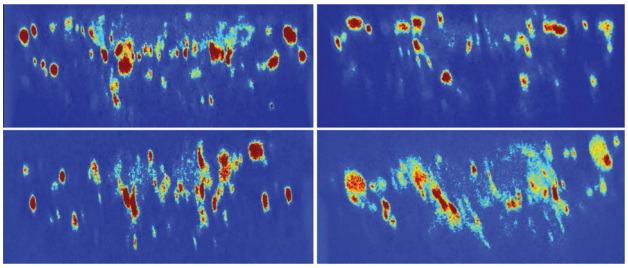

Fig. 2 Domain gap between nervous systems across subjects. Neural imaging data from four different animals in each corner. Images differ in terms of total brightness, the location of observed neurons, the number of visible neurons, and the shape and size of axons

data for epileptic patients (Peterson, 2021; Singh et al., 2021) along with the well-known H36M dataset (Ionescu et al., 2014) in which we treat the multiple views as independent domains. In all of the datasets, our ultimate goal is to interpret neural or video data so that one can generate latent representations that are useful for action recognition. Our method markedly improves across-subject action recognition in all datasets.

We hope our work will inspire the use and development of more general self-supervised neural feature extraction algorithms in neuroscience. These approaches promise to accelerate our understanding of how neural dynamics give rise to complex animal behaviors and can enable more robust neural decoding algorithms to be used in brain-machine interfaces.

2 Related Work

2.1 Neural Action Decoding

The ability to infer behavioral intentions from neural data, or neural decoding of behavior, is essential for the development of effective brain-machine interfaces and for closed-loop experimentation (Wen et al., 2021; Lau et al., 2021). Neural decoders can be used to increase the mobility of patients with disabilities (Collinger et al., 2018; Ganzer et al., 2020), or neuromuscular diseases (Utsumi et al., 2018), and can expand our understanding of how the nervous system works (Sani et al., 2018). However, most neural decoding methods require manual annotations of training data that are both tedious to acquire and error prone (Glaser et al., 2020; Lacourse et al., 2020; Segalin et al., 2021).

Existing self-supervised neural decoding methods (Wang et al., 2018; Kostas et al., 2021; Mohsenvand et al., 2020; Peterson et al., 2021) cannot be used on unlabeled subjects without action labels. A potential solution would be to use domain adaptation techniques to treat each new subject as a new domain. However, existing domain adaptation studies of neural decoding (Li et al., 2020; Farshchian et al., 2018) have focused on gradual domain shifts associated with slow changes in sensor measurements rather than the challenge of generalizing across individual subjects. In contrast to these methods, our approach is self-supervised and can generalize to unlabeled subjects at test time, without requiring action labels for new individuals.

2.2 Action Recognition

Contrastive learning has been extensively used on human motion sequences to perform action recognition using 3D pose data (Liu et al., 2020; Su et al., 2020; Lin et al., 2020) and video-based action understanding (Pan et al., 2021; Dave et al., 2021). Similarly, supervised and unsupervised action recognition approaches have been used on animal datasets on pose or RGB modalities (Sun et al., 2021; Eyjolfsdottir, 2017; Eyjolfsdottir et al., 2014, 2017; Bohnslav et al., 2021; Wiltschko et al., 2015). However, a barrier to using these tools in neuroscience is that the statistics of our neural data—the locations and sizes of cells—and behavioral data—body part lengths and limb ranges of motion—can be very different from animal to animal, creating a large domain gap.

In theory, there are multimodal domain adaptation methods for action recognition that could deal with this gap (Munro & Damen, 2020; Chen et al., 2019; Xu et al., 2021). However, they assume supervision in the form of labeled source data. In most laboratory settings, where large amounts of data are collected and resources are limited, this is an impractical solution.

2.3 Representation Learning

Most efforts to derive a low dimensional representation of neural activity have used recurrent models (Nassar et al., 2019; Linderman et al., 2019, 2017), variational autoencoders (Gao et al., 2016; Pandarinath et al., 2018), and dynamical systems (Abbaspourazad et al., 2021; Shenoy & Kao, 2021). Video and pose data have previously been used to segment and cluster temporally related behavioral information (Sun et al., 2021; Segalin et al., 2020; Overman et al., 2021; Pereira et al., 2020; Johnson et al., 2020).

By contrast, there have been relatively few approaches developed to extract behavioral representations from neural imaging data (Batty et al., 2019; Sani et al., 2021; Glaser et al., 2020). Most have focused on identifying simple relationships between these two modalities using simple supervised methods, such as correlation analysis, generalized linear models (Robie et al., 2017; Musall et al., 2019; Stringer et al., 2019), or regressive methods (Batty et al., 2019). We present a joint modeling of motion capture and neural modalities to fully extract behavioral information from neural data using a self-supervised learning technique.

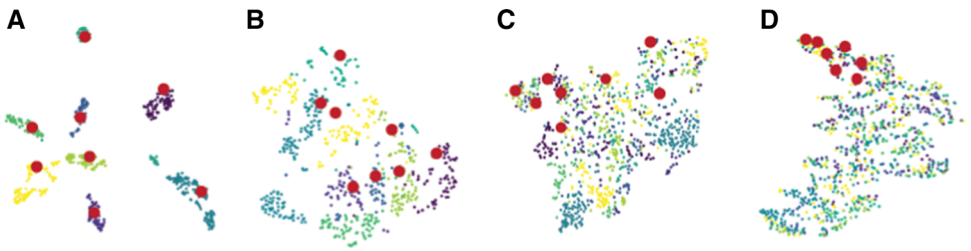

Fig. 3 t-SNE plots of the neural data. Each color denotes a different fly. The red dots are embeddings of the same action label in two different subjects. **A** Raw neural data. **B** SimCLR (Chen et al., 2020) representation, **C** Domain adaptation using a two-layer MLP discriminator and a Gradient Reversal Layer. **D** Ours. The identity of the animals is discarded and the semantic structure is preserved better than the previous methods, as similar same actions are positioned similarly, irrespective of subject identity

2.4 Pose Estimation

In order to utilize the advances in large behavioral recordings, recent efforts have made it possible to perform markerless predictions of 2D poses on animals using mostly deep learning (Pereira et al., 2020; Wu et al., 2020; Bala et al., 2020; Graving et al., 2019; Li et al., 2020). 2D animal poses can be converted into 3D animal poses using multi-view stereo systems or using lifting methods (Karashchuk et al., 2021; Günel et al., 2019; Gosztolai et al., 2021; Pedersen et al., 2020). Similarly, multi-animal tracking and pose estimation can be achieved using deep learning (Koger et al., 2022; Walter & Couzin, 2021). At the same time, realistic animal models have been built for downstream applications such as extracting 3D shape and texture from images, for animals such as mice, zebras, and elephants (Kulkarni et al., 2020; Sanakoyeu et al., 2020; Lobato-Rios et al., 2021; Bolaños et al., 2021).

2.5 Mixup Training

Mixup regularization was first proposed as a way to learn continuous latent spaces and to improve generalization for supervised learning (Berthelot et al., 2019; Verma et al., 2019; Zhang et al., 2018). Several previous studies have used a Mixup strategy to generate new positive pairs in contrastive learning (Shen et al., 2022; Lee et al., 2021). Mixup has rarely been used for domain adaptation. Recent examples include temporal background mixing (Sahoo et al., 2021), prediction smoothing across domains (Mao et al., 2019), and training better discriminators on uni-modal datasets (Sahoo et al., 2020). Our Mixup strategy can be regarded as a multi-modal extension of the previous approaches (Sahoo et al., 2021; Zhang et al., 2018) where per-frame feature-level stochastic Mixup between domains was performed to explore shared space and to hide identity information. Unlike these approaches, we explicitly condition the sampling procedure on the input data. We demonstrate that this approach helps to learn domain-invariant neural features.

In this work, we propose a new action recognition system by learning joint neural-behavioral representations using multi-modal pre-training. We learn these joint representations together with a novel set of augmentation strategies. Our method performs action classification without requiring action labels in the target domain. We show that our method outperforms previous neural action decoding work on three different datasets. We hope our method will accelerate our understanding of how neural dynamics give rise to complex animal behaviors and can enable more robust neural decoding algorithms to be used in brain-machine interfaces.

3 Approach

Our ultimate goal is to interpret neural data so that, given a neural image, one can generate latent representations that are useful for downstream tasks. This is challenging due to the wide domain-gap in neural representations between different subjects (Fig. 2). Hence, we aim to leverage self-supervised learning techniques to derive rich features that, once trained, could be used on downstream tasks including action recognition to predict the behaviors of unlabeled subjects.

Our data is composed of set of neural images synchronized with behavioral data, where we do not know where each action starts and ends. We leveraged contrastive learning to generate latent vectors from both modalities such that their mutual information would be maximized and therefore describe the same underlying action. However, this is insufficient to address the domain-gap between subjects (Fig. 3B). To do so, we implement an across-domain mixing strategy: We replace the original pose or neural data of an animal with mix of another set of animals from the same dataset, for which there is a high degree of 3D pose similarity at each given instance in time. Unlike behavioral data, neural data has unique properties. Neural calcium data contains information about previous actions because it decays slowly across time and it involves limited spatial resolution. To teach our model the invariance of these artifacts of neural data,

we propose two data augmentation techniques: (i) Neural Calcium augmentation - given a sequence of neural data, we apply an exponentially decaying neural snapshot to the sequence, which imitates the decaying impact of previous actions, (ii) Neural Noise augmentation - to make the model more robust to noise, we applied an augmentation which merges a sequence of neural data with another randomly sampled neural sequence using a coefficient.

Together, these augmentations enable a self-supervised approach to (i) bridge the domain gap between subjects allowing testing on unlabeled ones, and (ii) imitate the temporal and spatial properties of neural data, diversifying it and making it more robust to noise. In the following section, we describe these steps in more detail.

3.1 Problem Definition

We assume a paired set of data $\mathcal{D}_s = \left\{\left(\mathbf{b}_\mathbf{i}^s, \mathbf{n}_\mathbf{i}^s\right)\right\}_{i=1}^{n_s}$, where $\mathbf{b}_\mathbf{i}^s$ and $\mathbf{n}_\mathbf{i}^s$ represent behavioral and neural information respectively, with n_s being the number of samples for subject $s \in \mathcal{S}$. We quantify behavioral information $\mathbf{b}_\mathbf{i}^s$ as a set of 3D poses $\mathbf{b}_k^s$ for each frame $k \in \mathbf{i}$ taken of subject s, and neural information $\mathbf{n}_\mathbf{i}^s$ as a set of two-photon microscope images $\mathbf{n}_k^s$, for all frames $k \in \mathbf{i}$ capturing the activity of neurons. The data is captured such that the two modalities are always synchronized (paired) without human intervention, and therefore describe the same set of events. Our goal is to learn an unsupervised parameterized image encoder function f_n, that maps a set of neural images $\mathbf{n}_\mathbf{i}^s$ to a low-dimensional representation. We aim for our learned representation to be representative of the underlying action label, while being agnostic to both modality and the identity. We assume that we are not given action labels during pre-training. Also note that we do not know at which point in the captured data an action starts and ends. We just have a series of unknown actions performed by different subjects.

3.2 Contrastive Representation Learning

For each input pair $\left(\mathbf{b}_\mathbf{i}^s, \mathbf{n}_\mathbf{i}^s\right)$, we first draw a random augmented version $(\tilde{\mathbf{b}}_\mathbf{i}^s, \tilde{\mathbf{n}}_\mathbf{i}^s)$ with a sampled transformation function $t_n \sim \mathcal{T}_n$ and $t_b \sim \mathcal{T}_b$, where $\mathcal{T}_n$ and $\mathcal{T}_b$ represent a family of stochastic augmentation functions for behavioral and neural data, respectively, which are described in the following sections. Next, the encoder functions f_b and f_n transform the input data into low-dimensional vectors $\mathbf{h}_b$ and $\mathbf{h}_n$, followed by non-linear projection functions g_b and g_n, which further transform data into the vectors $\mathbf{z}_b$ and $\mathbf{z}_n$. For the behavioral modality, in order to facilitate mixing, we first transform augmented input data $\tilde{\mathbf{b}}_\mathbf{i}^s$ into $\mathbf{m}_\mathbf{i}^s$ using a shallow frame-wise MLP, as shown in Fig 4. For the neural modality, we instead directly apply mixing using $\mathbf{h}_n$, since frame-level mixing is not possible. We give the details of the mixing strategy in the next sections. During training, we sample a minibatch of N input pairs $\left(\mathbf{b}_\mathbf{i}^s, \mathbf{n}_\mathbf{i}^s\right)$, and train with the loss function

$$\mathcal{L}_{NCE}^{b\to n} = -\sum_{i=1}^{N} \log \frac{\exp\left(\left\langle \mathbf{z}_b^i, \mathbf{z}_n^i \right\rangle / \tau\right)}{\sum_{k=1}^{N} \exp\left(\left\langle \mathbf{z}_b^i, \mathbf{z}_n^k \right\rangle / \tau\right)} \tag{1}$$

where $\left\langle \mathbf{z}_b^i, \mathbf{z}_n^i \right\rangle$ is the cosine similarity between behavioral and neural modalities and $\tau \in \mathbb{R}^+$ is the temperature parameter. Intuitively, the loss function measures classification accuracy of a N-class classifier that tries to predict $\mathbf{z}_n^i$ given the true pair $\mathbf{z}_b^i$. To symmetrize the loss function with respect to the negative samples, we also define

$$\mathcal{L}_{NCE}^{n\to b} = -\sum_{i=1}^{N} \log \frac{\exp\left(\left\langle \mathbf{z}_b^i, \mathbf{z}_n^i \right\rangle / \tau\right)}{\sum_{k=1}^{N} \exp\left(\left\langle \mathbf{z}_b^k, \mathbf{z}_n^i \right\rangle / \tau\right)}. \tag{2}$$

We take the combined loss function to be $\mathcal{L}_{NCE} = \mathcal{L}_{NCE}^{b\to n} + \mathcal{L}_{NCE}^{n\to b}$, as in Zhang et al. (2020), Yuan et al. (2021). The loss function maximizes the mutual information between two modalities (van den Oord et al., 2019). Although standard contrastive learning bridges the gap between different modalities, it does not bridge the gap between different subjects (Fig. 3B). This is a fundamental challenge that we address in this work through augmentations described in the following section, which are part of the neural and behavioral family of augmentations $\mathcal{T}_n$ and $\mathcal{T}_b$.

3.2.1 Mixup Strategy

Given a set of consecutive 3D poses $\mathbf{b}_\mathbf{i}^s$ and their features $\mathbf{m}_\mathbf{i}^s$ calculated by a shallow MLP from augmented $\tilde{\mathbf{b}}_\mathbf{i}^s$, for each $k \in \mathbf{i}$, we stochastically replace $\mathbf{m}_k^s$ with a mix of its two pose neighbors, sampled from two subjects, in the set of domains $\mathcal{D}_\mathcal{S}$, where $\mathcal{S}$ is the set of all animals. To get one of the neighbors, we first uniformly sample a domain $\hat{s} \in \mathcal{S}$ and define a probability distribution $\mathbf{P}_{\mathbf{b}_k^s}^{\hat{s}}$ over the domain $\mathcal{D}_{\hat{s}}$ with respect to single 3D pose $\mathbf{b}_k^s$,

$$\mathbf{P}_{\mathbf{b}_k^s}^{\hat{s}}(\mathbf{b}_l^{\hat{s}}) = \frac{\exp(-\|\mathbf{b}_l^{\hat{s}} - \mathbf{b}_k^s\|_2)}{\sum_{\mathbf{b}_m^{\hat{s}} \in \mathcal{D}_{\hat{s}}} \exp(-\|\mathbf{b}_m^{\hat{s}} - \mathbf{b}_k^s\|_2)}. \tag{3}$$

We then sample from the above distribution and pass it through the shallow MLP, which yields $\mathbf{m}_l^{\hat{s}} \sim \mathbf{P}_{\mathbf{b}_k^s}^{\hat{s}}$. Notice that, although distribution is conditioned on the 3D pose $\mathbf{b}^s$, we sample back 3D pose features $\mathbf{m}^s$. In practice, we calculate the distribution $\mathbf{P}$ only over the approximate $\mathbf{N}$ nearest neighbors of $\mathbf{b}_k^s$, in order to speed up the implementation. We empirically set $\mathbf{N}$ to 128. Given two samples $\mathbf{m}_l^{\hat{s}}$ and $\mathbf{m}_j^{\bar{s}}$

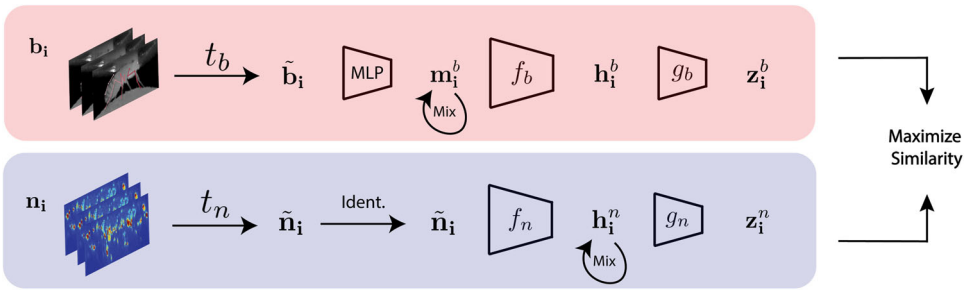

Fig. 4 Our approach to learning an effective representation of behaviors. First, we sample a synchronized set of behavioral and neural frames, $(\mathbf{b_i}, \mathbf{n_i})$. Then, we augment these data using randomly sampled augmentation functions t_b and t_n. Encoders f_b and f_n generate intermediate representations $\mathbf{h}^b$ and $\mathbf{h}^n$, which are then projected into $\mathbf{z}_b$ and $\mathbf{z}_t$ by two separate projection heads g_b and g_n. For the behavioral modality, we first apply a frame-wise MLP before f_b. We then apply mixing on $\mathbf{m}_\mathbf{i}^b$. For the neural modality, we apply mixing on $\mathbf{h}_\mathbf{i}^n$ without an MLP, since mixing cannot be done at frame level. We maximize the similarity between the two projections using an InfoNCE loss. At test time, the red branch and $\mathbf{h}_\mathbf{i}^n$ is used for neural decoding

from the above distribution from independent domains, we then return the mixed version of

$$\tilde{\mathbf{m}}_k^s = \lambda \mathbf{m}_l^{\hat{s}} + (1-\lambda)\mathbf{m}_j^{\bar{s}} \,. \tag{4}$$

We sample the mixing coefficient λ from the Beta distribution $\lambda \sim \text{Beta}(\alpha, \beta)$. Our Mixup strategy removes the identity information in the behavioral data without perturbing it to the extent that semantic action information is lost. Since each behavioral sample $\mathbf{m}_\mathbf{i}^s$ is composed of a set of 3D pose features, and each 3D pose feature $\mathbf{m}_k^s, \forall k \in \mathbf{i}$ is replaced with a feature of a random domain, the transformed sample $\tilde{\mathbf{m}}_\mathbf{i}^s$ is now composed of multiple domains. This forces the behavioral encoding function f_b to leave identity information out, therefore generalizing across multiple domains (Fig. 5).

Our Mixup augmentation is similar to the synonym replacement augmentation used in natural language processing (Wei & Zou, 2019), where randomly selected words in a sentence are replaced by their synonyms, therefore changing the syntactic form of the sentence without altering the semantics. Instead, we randomly replace each 3D pose in a motion sequence. To the best of our knowledge, we are the first to use frame-wise mix strategy in the context of time-series analysis or for domain adaptation that is conditioned on the input.

To keep mixing symmetric, we also mix the neural modality. To mix a set of neural features $\mathbf{h}_\mathbf{i}^s$, we take its behavioral pair $\mathbf{b}_\mathbf{i}^s$, and search for similar sets of poses in other domains, with the assumption that similar sets of poses describe the same action. Therefore, once similar behavioral data is found, their neural data can be mixed. Note that, unlike behavior mixing, we do not calculate the distribution on individual 3D pose $\mathbf{b}_k^s$, but instead on the whole set of behavioral data $\mathbf{b}_\mathbf{i}^s$, because similarity in a single pose does not necessarily imply similar actions and similar neural data. More formally, given the behavioral-neural pair $\left(\mathbf{b}_\mathbf{i}^s, \mathbf{n}_\mathbf{i}^s\right)$, we mix the neural modality features $\mathbf{h}_\mathbf{i}^s$ by sampling two new neural features $\mathbf{h}_\mathbf{j}^{\bar{s}}$ and $\mathbf{h}_\mathbf{l}^{\hat{s}}$ from distinct animals $\hat{s}$ and $\bar{s}$, using the probability distribution

$$\mathbf{P}_{\mathbf{n}_\mathbf{i}^s}^{\hat{s}}(\mathbf{b}_\mathbf{j}^{\hat{s}}) = \frac{\exp(-\|\mathbf{b}_\mathbf{j}^{\hat{s}} - \mathbf{b}_\mathbf{i}^s\|_2)}{\sum_{\mathbf{b}_\mathbf{m}^{\hat{s}} \in \mathcal{D}_{\hat{s}}} \exp(-\|\mathbf{b}_\mathbf{m}^{\hat{s}} - \mathbf{b}_\mathbf{i}^s\|_2)}, \tag{5}$$

and then we return

$$\tilde{\mathbf{h}}_k^s = \lambda \mathbf{h}_l^{\hat{s}} + (1-\lambda)\mathbf{h}_j^{\bar{s}} \,. \tag{6}$$

Similarly, first we sample the mixing coefficient λ from the Beta distribution $\lambda \sim \text{Beta}(\alpha, \beta)$. This yields new mixed neural feature $\tilde{\mathbf{h}}_\mathbf{i}^s$, where the augmented neural data comes from two different subjects in $\mathcal{S}$.

3.2.2 Neural Calcium Augmentation

Our neural data was obtained using two-photon microscopy and fluorescence calcium imaging. The resulting images are only a function of the underlying neural activity, and have temporal properties that differ from the true neural activity. For example, calcium signals from a neuron change much more slowly than the neuron's actual firing rate. Consequently, a single neural image $\mathbf{n}_t$ includes decaying information concerning neural activity from the recent past, and thus carries information about previous behaviors. This makes it harder to decode the current behavioral state.

We aimed to prevent this overlap of ongoing and previous actions. Specifically, we wanted to teach our network to be invariant with respect to past behavioral information by augmenting the set of possible past actions. To do this, we generated new data $\tilde{\mathbf{n}}_\mathbf{i}^s$, that included previous neural activity $\mathbf{n}_k^s$. To mimic calcium indicator decay dynamics, given a neural data sample $\mathbf{n}_\mathbf{i}^s$ of multiple frames, we sample a new

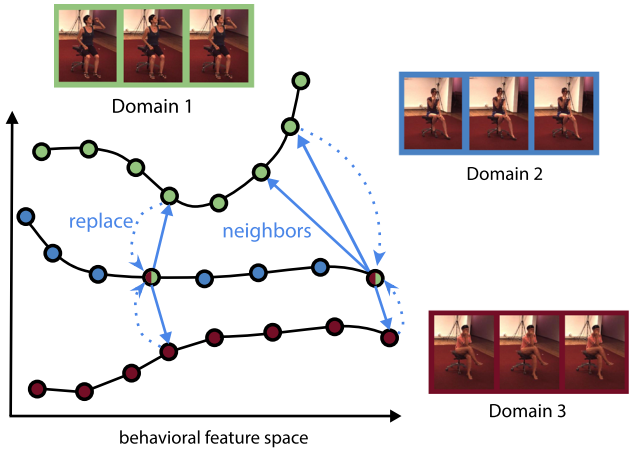

Fig. 5 Our Mixup strategy. Each 3D pose is processed by a pose-wise MLP to generate 3D pose features. Then, each 3D pose feature in the motion sequence of Domain 2 is randomly replaced with mix of two of its neighbors, from the set of domains $\hat{s} \in \mathcal{S}$, which includes Domains 1 and 3. The Mixup augmentation hides identity information, while keeping pose changes in the sequence minimal

neural frame $\mathbf{n}_k^s$ from the same domain, where $k \notin \mathbf{i}$. We then convolve $\mathbf{n}_k^s$ with the temporally decaying calcium convolutional kernel $\mathcal{K}$, therefore creating a set of images from a single frame $\mathbf{n}_k^s$, which we then add back to the original data sample $\mathbf{n}_\mathbf{i}^s$. This results in $\tilde{\mathbf{n}}_\mathbf{i}^s = \mathbf{n}_\mathbf{i}^s + \mathcal{K} * \mathbf{n}_k^s$ where $*$ denotes the convolutional operation. In the Supplementary Material, we explain calcium dynamics and our calculation of the kernel $\mathcal{K}$ in more detail.

3.2.3 Neural Noise Augmentation

Two-photon microscopy images often include multiple neural signals combined within a single pixel. This is due to the the fact that multiple axons can be present in a small tissue volume that is below the spatial resolution of the microscope. To mimic this noise-adding effect, given a neural image $\mathbf{n}_\mathbf{i}^s$, we randomly sample a set of frames $\mathbf{n}_\mathbf{k}^{\hat{s}}$, from a random domain $\hat{s}$. We then return the blend of these two videos, $\tilde{\mathbf{n}}_\mathbf{i}^s = \mathbf{n}_\mathbf{i}^s + \alpha \mathbf{n}_\mathbf{k}^{\hat{s}}$, to mix and hide the behavioral information. Unlike the CutMix (Yun et al., 2019) augmentations used for supervised training, we apply the augmentation in an unsupervised setup to make the model more robust to noise. We sample a random α for the entire set of samples in $\mathbf{n}_\mathbf{i}^s$.

4 Experiments

We test our method on three datasets. In this section, we describe these datasets, the set of baselines against which we compare our model, and finally the quantitative comparison of all models.

4.1 Datasets

We ran most of our experiments on a large dataset of fly neural and behavioral recordings that we acquired and describe below, which we called MC2P. To demonstrate our method's ability to generalize, we also adapted it to run on another multimodal dataset that features neural ECoG recordings and markerless motion capture (Peterson, 2021; Singh et al., 2021), as well as the well known H36M human motion dataset (Ionescu et al., 2014).

4.1.1 MC2P

Since there was no available neural-behavioral dataset with a rich variety of spontaneous behaviors from multiple individuals, we acquired our own dataset that we name *Motion Capture and Two-photon Dataset (MC2P)*. We will release this dataset publicly. MC2P features data acquired from tethered behaving adult flies, *Drosophila melanogaster* (Fig. 1), It includes:

1. Infrared video sequences of the fly acquired using six synchronized and calibrated infrared cameras forming a ring with the animal at its center. The images are 480×960 pixels in size and recorded at 100 fps.
2. Neural activity imaging obtained from the axons of descending neurons that pass from the brain to fly's ventral nerve cord (motor system) and drive actions. The neural images are 480×736 pixels in size and recorded at 16 fps using a two-photon microscope (Chen et al., 2018) that measures the calcium influx which is a proxy for the neuron's actual firing rate.

We recorded 40 animals over 364 trials, resulting in 20.7 hours of recordings with 7,480,000 behavioral images and 1,197,025 neural images. We provide additional details and examples in the Supplementary Material. We give an example video of synchronized behavioral and neural modalities in Supplementary Videos 1 and 2.

To obtain quantitative behavioral data from video sequences, we extracted 3D poses expressed in terms of the 3D coordinates of 38 keypoints (Günel et al., 2019). We provide an example of detected poses and motion capture in Supplementary Videos 3 and 4. For validation purposes, we manually annotated a subset of frames using eight behavioral labels: *forward walking*, *pushing*, *hindleg grooming*, *abdominal grooming*, *rest*, *foreleg grooming*, *antenna grooming*, and *eye grooming*. We provide an example of behavioral annotations in Supplementary Video 5. To keep the experiments consistent, we always paired 32 frames of neural data with 8 frames of behavioral data.

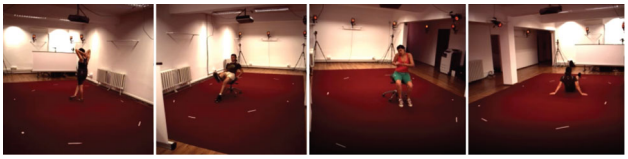

Fig. 6 Domain Gap in the H3.6M dataset. Similar to the domain gap across nervous systems, RGB images show a significant domain gap when the camera angle changes across individuals. We guide action recognition across cameras in RGB images using 3D poses and behavioral mixing

4.1.2 ECoG Dataset

(Peterson, 2021; Singh et al., 2021): This dataset was recorded from epilepsy patients over a period of 7-9 days. Each patient had 90 electrodes implanted under their skull. The data comprises human neural Electrocorticography (ECoG) recordings and markerless motion capture of upper-body 2D poses. The dataset is labeled to indicate periods of voluntary spontaneous motions, or rest. As for two-photon images in flies, ECoG recordings show a significant domain gap across individual subjects. We applied our multi-modal contrastive learning approach on ECoG and 2D pose data along with mixing-augmentation. Then, we applied an across-subject benchmark in which we do action recognition on a new subject without known action labels (Fig. 6).

4.1.3 H3.6M

H3.6M is a multi-view motion capture dataset that is not inherently multimodal. However, to test our approach in a very different context than the other two cases, we treated the videos acquired by different camera angles as belonging to separate domains. Since videos are tied to 3D poses, we used these two modalities and applied mixing augmentation together with multimodal contrastive learning to reduce the domain gap across individuals. Then, we evaluated the learned representations by performing action recognition on a camera angle that we do not have action labels for. This simulates our across-subject benchmark used in the MC2P dataset. For each experiment we selected three actions, which can be classified without examining large window sizes. We give additional details in the Supplementary Material.

4.2 Baselines

We evaluated our method using two supervised baselines, Neural Linear and Neural MLP. These directly predict action labels from neural data without any self-supervised pretraining using cross-entropy loss. We do not use any post-processing or smoothing after any of our baselines. We also compared our approach to three regression methods that attempt to regress behavioral data from neural data, which is a common neural decoding technique. These include a recent neural decoding algorithm, BehaveNet (Batty et al., 2019), as well as to two other regression baselines with recurrent and convolutional approaches: Regression (Recurrent) and Regression (Convolution). In addition, we compare our approach to recent self-supervised representation learning methods, including SeqCLR (Mohsenvand et al., 2020) and SimCLR (Chen et al., 2020). We also combine convolutional regression-based method (Reg. (Conv)) or the self-supervised learning algorithm SimCLR with the common domain adaptation techniques Gradient Reversal Layer (GRL) (Ganin & Lempitsky, 2015), or Mean Maximum Discrepancy (Gretton et al., 2006). This yields four domain adaptation models. Finally, we apply a recent multi-modal domain adaptation network for action recognition, MM-SADA (Munro & Damen, 2020) on MC2P dataset. For all of these methods, we used the same backbone architecture. We describe the backbone architecture in more detail in the Supplementary Material. We describe the baselines in more detail in following:

4.2.1 Supervised

A feedforward network trained with manually annotated action labels using cross-entropy loss, having neural data as input. We discarded datapoints that did not have associated behavioral labels. For the MLP baseline, we trained a simple three layer MLP with a hidden layer size of 128 neurons with ReLU activation and without batch normalization.

4.2.2 Regression (Convolutional)

A fully-convolutional feedforward network trained with MSE loss for behavioral reconstruction task, given the set of neural images. To keep the architectures consistent with the other methods, the average pooling is followed by a projection layer, which is used as the final representation of this model.

4.2.3 Regression (Recurrent)

This is similar to the one above but the last projection network was replaced with a two-layer GRU module. The GRU module takes as an input the fixed representation of neural images. At each time step, the GRU module predicts a single 3D pose with a total of eight steps to predict the eight poses associated with an input neural image. This model is trained with an MSE loss. We take the input of the GRU module as the final representation of neural encoder.

4.2.4 BehaveNet

This uses a discrete autoregressive hidden Markov model (ARHMM) to decompose 3D motion information into discrete "behavioral syllables" (Batty et al., 2019). As in the regression baseline, the neural information is used to predict the posterior probability of observing each discrete syllable. Unlike the original method, we used 3D poses instead of RGB videos as targets. We skipped compressing the behavioral data using a convolutional autoencoder because, unlike RGB videos, 3D poses are already low-dimensional.

4.2.5 SimCLR

We trained the original SimCLR module without the calcium imaging data and mixing augmentations (Chen et al., 2020). As in our approach, we took the features before the projection layer as the final representation.

4.2.6 Gradient Reversal Layer (GRL)

Together with the contrastive loss, we trained a two-layer MLP domain discriminator per modality, D_b and D_n, which estimates the domain of the neural and behavioral representations (Ganin & Lempitsky, 2015). Discriminators were trained by minimizing

$$\mathcal{L}_D = \sum\nolimits_{x\in\{\mathbf{b},\mathbf{n}\}} -d \log\left(D_m\left(f_m(x)\right)\right) \tag{7}$$

where d is the one-hot identity vector. Gradient Reversal layer is inserted before the projection layer. Given the reversed gradients, the neural and behavioral encoders f_n and f_b learn to fool the discriminator and outputs invariant representations across domains, hence acting as a domain adaptation module. We kept the hyperparameters of the discriminator the same as in previous work (Munro & Damen, 2020). We froze the weights of the discriminator for the first 10 epochs, and trained only the $\mathcal{L}_{NCE}$. We trained the network using both loss functions, $\mathcal{L}_{NCE} + \lambda_D \mathcal{L}_D$, for the remainder of training. We set the hyperparameters λ_D to 10 empirically.

4.2.7 Maximum Mean Discrepancy (MMD)

We replaced adversarial loss in GRL baseline with a statistical test that minimizes the distributional discrepancy from different domains (Gretton et al., 2006).

4.2.8 MM-SADA

A recent multi-modal domain adaptation model for action recognition that minimizes cross-entropy loss on target labels, adverserial loss for domain adaptation, and contrastive losses to maximize consistency between multiple modalities (Munro & Damen, 2020). As we do not assume any action labels during the contrastive training phase, we removed the cross-entropy loss.

4.2.9 SeqCLR

This approach learns a uni-modal self-supervised contrastive model (Mohsenvand et al., 2020). Hence, we only apply it to the neural imaging data, without using the behavioral modality. As this method was previously applied on datasets with Electroencephalography (ECoG) imaging technique, we removed ECoG specific augmentations.

4.2.10 Maximum Mean Discrepancy (MMD)

We replaced adversarial loss in GRL baseline with a statistical test to minimize the distributional discrepancy from different domains (Gretton et al., 2006). Similar to previous work, we applied MMD only on the representations before the projection layer independently on both modalities (Munro & Damen, 2020; Kang et al., 2020). Similar to the GLR baseline, we first trained 10 epochs only using the contrastive loss, and trained using the combined losses $\mathcal{L}_{NCE}+\lambda_{MMD}\mathcal{L}_{MMD}$ for the remainder. We set the hyperparameters λ_{MMD} as 1 empirically. For the domain adaptation methods GRL and MMD, we reformulated the denominator of the contrastive loss function. Given a domain function dom which gives the domain of the data sample, we replaced one side of L_{NCE} in Eq. 1 with,

$$\log \frac{\exp\left(\left\langle \mathbf{z}_b^i, \mathbf{z}_n^i \right\rangle / \tau\right)}{\sum_{k=1}^{N} \mathbf{1}_{[dom(i)=dom(k)]} \exp\left(\left\langle \mathbf{z}_b^i, \mathbf{z}_n^k \right\rangle / \tau\right)}, \tag{8}$$

where selective negative sampling prevents the formation of trivial negative pairs across domains, therefore making it easier to merge multiple domains. Negative pairs formed during contrastive learning try to push away inter-domain pairs, whereas domain adaptation methods try to merge multiple domains to close the domain gap. We found that the training of contrastive and domain adaptation losses together could be quite unstable, unless the above changes were made to the contrastive loss function.

4.3 Benchmarks

Since our goal is to create useful representations of neural images in an self-supervised way, we focused on single- and across-subject action recognition. Specifically, we trained our neural decoder f_n along with the others without using any action labels. Then, freezing the neural encoder parameters, we trained a linear model on the encoded features, which

is an evaluation protocol widely used in the field (Chen et al., 2020; Lin et al., 2020; He et al., 2020; Dave et al., 2021). We used either half or all action labels. We mention the specifics of the train-test split in the Supplementary Material.

4.3.1 Single-Subject Action Recognition

For each subject, we trained and tested a simple linear classifier *independently* on the learned representations to predict action labels. We assume that we are given action labels on the subject we are testing. In Table 1 we report aggregated results.

4.3.2 Across-Subject Action Recognition

We trained linear classifiers on N-1 subjects simultaneously and tested on the left-out one. Therefore, we assume we do not have action labels for the target subject. We repeated the experiment for each individual and report the mean accuracy in Tables 1 and 2.

4.3.3 Identity Recognition

As a sanity check, we attempted to classify subject identity among the individuals given the learned representations. We again used a linear classifier to test the domain invariance of the learned representations. In the case that the learned representations are domain (subject) invariant, we expect that the linear classifier will not be able to detect the domain of the representations, resulting in a lower identity recognition accuracy. Identity recognition results are rerported in Tables 1 and 2.

5 Results

5.1 Single-Subject Action Recognition on M2CP

For the Single-Subject baseline, joint modeling of common latent space out-performed supervised models by a large margin, even when the linear classifier was trained on the action labels of the tested animal. Our mixing and neural augmentations resulted in an accuracy boost when compared with a simple contrastive learning method, SimCLR (Chen et al., 2020). Although regression-based methods can extract behavioral information from the neural data, they do not produce discriminative features. When combined with the proposed set of augmentations, our method performs better than previous neural decoding models because it extracts richer features thanks to a better self-supervised pretraining step. Domain adaptation techniques do not result in a significant difference in the single-subject baseline; the domain gap in a single animal is smaller than between animals.

5.2 Across-Subject Action Recognition on M2CP

We show that supervised models do not generalize across animals, because each nervous system is unique. Before using the proposed augmentations, the contrastive method SimCLR performed worse than convolutional and recurrent regression-based methods including the current state-of-art BehaveNet (Batty et al., 2019). This was due to large domain gap between animals in the latent embeddings (Fig. 3B). Although the domain adaptation methods MMD (Maximum Mean Discrepancy) and GRL (Gradient Reversal Layer) close the domain gap when used with contrastive learning, they do not position semantically similar points near one another (Fig. 3C). As a result, domain adaptation-based methods do not result in significant improvements in the across-subject action recognition task. Although regression-based methods suffer less from the domain gap problem, they do not produce representations that are as discriminative as contrastive learning-based methods. Our proposed set of augmentations and strategies close the domain gap, while improving the action recognition baseline for self-supervised methods, for both single-subject and across-subject tasks (Fig. 3D).

5.3 Action Recognition on ECoG Motion versus Rest

As shown at the bottom of Table 2, our approach significantly lowers the identity information in ECoG embeddings, while significantly increasing across-subject action recognition accuracy compared to the regression and multi-modal SimCLR baselines. Low supervised accuracy confirms a strong domain gap across individuals. Note that uni-modal contrastive modeling of ECoG recordings (SimCLR (ECoG)) does not yield strong across-subject action classification accuracy because uni-modal modeling cannot deal with the large domain gap in the learned representations.

5.4 Human Action Recognition on H3.6M

We observe in Table 2 that, similar to the previous datasets, the low performance of the supervised baseline and the uni-modal modeling of RGB images (SimCLR (RGB)) are due to the domain-gap in the across-subject benchmark. This observation is confirmed by the high identity recognition of these models. Our mixing strategy strongly improves compared to the regression and multi-modal contrastive (SimCLR) baselines. Similar to the previous datasets, uni-modal contrastive training cannot generalize across subjects, due to the large domain gap.

Table 1 Action recognition accuracy on MC2P dataset

Tasks →	Single-Subject ↑		Across-Subject ↑		Identity Recog. ↓		Pose
Percentage of Data →	0.5	1.0	0.5	1.0	0.5	1.0	
Random Guess	16.6	16.6	16.6	16.6	12.5	12.5	☐
Neural (Linear)	29.3	32.5	18.4	18.4	100.0	100.0	☐
Neural (MLP)	–	–	18.4	18.4	100.0	100.0	☐
SeqCLR (Mohsenvand et al., 2020)	39.5	42.1	21.9	28.4	93.0	96.5	☐
Ours (Neural Only)	42.0	44.8	21.3	30.6	94.1	96.8	☐
SimCLR (Chen et al., 2020)	54.3	57.6	46.9	50.6	69.9	80.3	☑
Regression (Recurrent)	53.6	59.7	49.4	51.2	89.5	91.8	☑
Regression (Convolution)	52.6	59.6	50.6	55.8	88.7	92.5	☑
BehaveNet (Batty et al., 2019)	54.6	60.2	50.5	56.8	80.2	83.4	☑
SimCLR (Chen et al., 2020) + MMD (Gretton et al., 2006)	53.6	57.8	50.1	53.1	18.4	21.2	☑
SimCLR (Chen et al., 2020) + GRL (Ganin & Lempitsky, 2015)	53.5	56.3	49.9	52.3	16.7	19.1	☑
Reg. (Conv.) + MMD (Gretton et al., 2006)	54.5	60.7	52.6	55.4	18.2	19.5	☑
Reg. (Conv.) + GRL (Ganin & Lempitsky, 2015)	55.5	60.2	51.8	55.7	17.2	17.3	☑
MM-SADA (Munro & Damen, 2020)	53.1	56.2	49.2	52.1	13.8	15.2	☑
Ours	**57.6**	**63.1**	**54.8**	**61.5**	**13.2**	**13.6**	☑

Bold values denote the best method for each category

Single- and Across-Subject action recognition results on the MC2P dataset. Neural MLP results for the single-subject task are removed because single subjects often do not have enough labels for every action. Smaller numbers are better for Identity Recognition. Our method performs better than previous neural decoding methods and other self-supervised learning based pre-training methods in all benchmarks, while at the same time closing the domain gap between animals, as shown by the identity recognition task. The last column specifies whether the method has access to motion capture information during training

Table 2 Action recognition accuracy on H36M and ECoG dataset

Dataset	Tasks → % of Data →	A.S. 0.5	A.S. 1.0	I.R. 1.0
H3.6M Walking, Sitting, Posing	Random Gu.	33.0	33.0	33.0
	Supervised	46.6	48.3	100.0
	SimCLR (RGB)	33.2	33.5	99.5
	SimCLR	53.3	55.7	99.2
	Regression (Conv.)	65.2	68.8	68.4
	Ours	72.4	73.6	42.3
H3.6M Walking, Directions, Eating	Random Gu.	33.3	33.3	33.3
	Supervised	31.2	30.9	100.0
	SimCLR (RGB)	34.6	34.4	100.0
	SimCLR	52.3	53.2	94.8
	Regression (Conv.)	44.8	48.7	62.1
	Ours	63.2	68.3	44.8
ECoG Moving, Rest	Random Gu.	50.0	50.0	33.3
	Supervised	54.2	53.8	100.0
	SimCLR (ECoG)	52.3	55.1	98.0
	SimCLR	64.6	72.1	81.1
	Regression (Conv.)	64.1	71.8	74.3
	Ours	75.8	81.9	53.0

Across-subject (A.S.) and identity recognition (I.R.) results on H3.6M dataset (Ionescu et al., 2014) using RGB and 3D pose, and on ECoG Move vs Rest (Peterson, 2021) using neural ECoG recordings and 2D pose. For Ours, we remove calcium imaging specific augmentations and only use mixing strategy. Mixing strategy closes the domain gap for the contrastive learning and strongly improves across-subject action recognition on both datasets

Table 3 Ablation on effects of different augmentations

Method	Single Subj.↑	Across Subj.↑	Identity Recog. ↓
w/ Mixing Strategy	▲ + 2.7	▲ + 7.9	▼ -63.0
+ w/ Calcium Augmentation	▲ + 2.1	▲ + 2.7	■ +1.2
+ w/ N. Noise Augmentation	▲ + 1.1	▲ + 1.2	■ -0.8

Showing the effect of different augmentations on single-subject, across-subject and identity recognition benchmarks on MC2P Dataset, when compared to simple contrastive model

Table 4 Ablation on neural preprocessing

Method	Single Subj.↑	Across Subj.↑	Identity Recog. ↓
CaImAn	56.8	56.1	17.0
Ours	63.1	61.5	13.6

Comparing standart neural processing library *CaImAn* with our augmentations on MC2P Dataset

5.5 Ablation Study

We compare the individual contributions of different augmentations proposed in our method. We report these results in Table 3. We observe that all augmentations contribute to single- and across-subject benchmarks. Our mixing augmentation strongly affects the across-subject benchmark, while at the same time greatly decreasing the domain gap, as quantified by the identity recognition result. Other augmentations have minimal effects on the domain gap, as they only slightly affect the identity recognition benchmark (Tables 4 and 5).

We compare our neural augmentations to standart neural preprocessing approaches commonly used in neuroscience. To compare, we use the state-of-art neural preprocessing library *CaImAn* (Giovannucci et al., 2019). *CaImAn* requires the tuning of 25 parameters for spike inference. Running a single-set of parameters took 5h of processing. As shown in Tab. 4, this algorithm produced worse results, likely due to errors in ROI detection and spike inference. Because our method can be run on the raw data, without requiring ROI detection and spike inference, it removes the burden of an extensive hyperparameter search and unnecessarily long computational times. Thus, we believe that our augmentations and model are more general and useful for the community. Lastly, we performed ablation experiment on mixing individual modalities and report the results in Tab. 5. Mixing both modalities results in the best scores. However, when mixed alone, the behavioral modality performs superior as it more effectively hides subject identity information, since it is mixed in pose level instead of window level.

Table 5 Ablation on mixing of different modalities on MC2P dataset

Method	Single Subj.↑	Across Subj.↑	Identity Recog. ↓
No Mix.	59.1	51.2	81.7
N. Mix.	60.4	58.9	36.8
B. Mix.	61.0	60.2	25.2
N. + B. Mix.	63.1	61.5	13.6

Showing the effect of mixing different modalities on MC2P dataset

6 Conclusion

We have introduced an self-supervised neural action representation framework for neural imaging and behavioral videography data. We extended previous methods by incorporating a new mixing based domain adaptation technique which we have shown to be useful on three very different multimodal datasets, together with a set of domain-specific neural augmentations. Two of these datasets are publicly available. We created the third dataset, which we call MC2P, by recording video and neural data for *Drosophila melanogaster* and will release it publicly to speed-up the development of self-supervised methods in neuroscience (Aymanns et al., 2022). We hope our work will help the development of effective brain machine interface and neural decoding algorithms. In future work, we plan to disentangle remaining long-term non-behavioral information that has a global effect on neural data, such as hunger or thirst, and test our method on different neural recording modalities. As a potential negative impact, we assume that once neural data is taken without consent, our method can be used to extract private information.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11263-022-01713-6.

Funding Open access funding provided by EPFL Lausanne

Appendix for Overcoming the Domain Gap in Neural Action Representations

Human Actions

We apply multi-modal contrastive learning on windows of time series and RGB videos. We make the analogy that, similar to the neural data, RGB videos from different view angles show a domain gap although they are tied to the same 3D pose. Therefore, to test our method, we select three individuals with different camera angles where all actors perform the same three actions. We test domain adaptation using the Across-Subject benchmark, where we train our linear action classifier on labels of one individual and test it on the others. We repeat the same experiment three times and report the mean results. We show the results of Across-Subject and Identity Recognition in Table 2.

For preprocessing, we remove global translation and rotation from 3D poses by subtracting the root joint and then rotating the skeletons to point in the same direction. We use resnet18 for the RGB encoder and a 4 layer convolutional network for the 3D pose encoder. We use S1, S5 and S7 and all their behaviors for training, except for the three behaviors which we used for testing. For each number, we report three-fold cross-validation results (Fig. 7).

Dataset Details

Dataset Collection

Here we provide a more detailed technical explanation of the experimental dataset. Transgenic female *Drosophila melanogaster* flies aged 2-4 days post-eclosion were selected for experiments. They were raised on a 12h:12h day, night light cycle and recorded in either the morning or late afternoon Zeitgeber time. Flies expressed both GCaMP6s and tdTomato in all brain neurons as delineated by otd-Gal4 expression, (; $\frac{Otd-nls:FLPo(attP40)}{P20XUAS-IVS-GCaMP6sattP40}$; $\frac{R57C10-GAL4,tub>GAL80>}{Pw[+mC]=UAS-tdTom.S3}$.) The fluorescence of GCaMP6s proteins within the neuron increases when it binds to calcium. There is an increase in intracellular calcium when neurons become active and fire action potentials. Due to the relatively slow release (as opposed to binding) of calcium by GCaMP6s molecules, the signal decays exponentially. We also expressed the red fluorescent protein, tdTomato, in the same neurons as an anatomical fiduciary to be used for neural

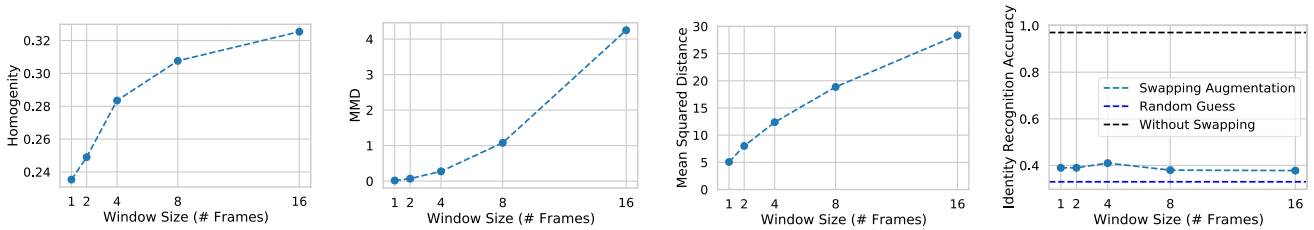

Fig. 7 Changing window size for the mixing of behavioral modality on the MC2P dataset. Statistics of the behavioral modality as a function of changing the window size. Decreasing the window size increases clustering homogeneity and Mean Maximum Discrepancy (MMD) when applied to the raw data, therefore suggesting higher quality mixing in individual poses instead of sequences of poses. mixing augmentation with a smaller window size lowers the degree of perturbation, quantified by Mean Squared Distance. However, identity recognition accuracy does not change considerably when mixing is done with different window sizes

data registration. This compensates for image deformations and translations during animal movements. We recorded neural data using a two-photon microscope (ThorLabs, Germany; Bergamo2) by scanning the cervical connective. This neural tissue serves as a conduit between the brain and ventral nerve cord (VNC) (Chen et al., 2018). The brain-only GCaMP6s expression pattern in combination with restrictions of recording to the cervical connective allowed us to record a large population of descending neuron axons while also being certain that none of the axons arose from ascending neurons in the VNC. Because descending neurons are expected to drive ongoing actions (Cande et al., 2018), this imaging approach has the added benefit of ensuring that the imaged cells should, in principle, relate to paired behavioral data.

To synchronize two-photon images and RGB-signals, which are acquired at different sampling rates, we use BNC 2110 terminal block (National Instrument, USA) and ThorSync software (Thorlabs, USA). We record both modalities using our custom setup Fig. 8. We then use sampling timestamps as references to align data; each neural frame is associated with the behavioral frame with the closest timestamps.

To create behavioral modality for the MC2P dataset, we used the off-the-shelf DeepFly3D network with pre-trained weights (Günel et al., 2019). The input to DeepFly3D is video data from our six infrared cameras. These images are then used to identify the 3D positions of 38 landmarks per animal using DeepFly3D: (i) five on each limb - the thorax-coxa, coxa-femur, femur-tibia, and tibia-tarsus joints as well as the pretarsus, (ii) six on the abdomen - three on each side, and (iii) one on each antenna. DeepFly3D detects arbitrary points on the fly's body and relies on bundle adjustment to simultaneously assign 3D locations to these points and to estimate the positions and orientations of each camera. DeepFly3D is trained on 37,000 frames which were created automatically using an active learning system, and another 3,000 manual annotations. DeepFly3D achieves a Root Mean Square Error (RMSE) of 13.9 pixels. We compared the reported DeepFly3D error on their original dataset with our own predictions in our MC2P dataset. We observed resulting error is around 21.1 pixels. We believe the larger error is due to the different experimental settings, and slightly different illumination conditions. Although the error is quantitatively larger, visually we found little difference from the original performance. Similar to the previous reports, we have found extremities (tarsus tip) exhibited larger errors than the other joints, perhaps due to occlusions from the setup, and higher variance overall. For the action annotations, we have used two sets of annotators. We have not observed a significant annotation difference between the annotators. In general, the event starts and ends only differ by less than 5.4 frames between the two annotators on average. This is much smaller than the average length of action, 116 frames. We then continued to label actions using a single annotator. We believe the small error rate is due to the relatively stereotyped behavior repertoire of *Drosohpila*, which makes it much easier to label actions.

Behavioral Pre-processing

For the MC2P dataset, we register each 3D pose into a canonical coordinate system using Procrustes's analysis and normalize limb-lengths across subjects. Since each animal is tethered under the two-photon microscope, they do not change their rotation during the experiment. For each of the six legs, we set the body-coxa locations so that relative sizes of the animals do not reflected in the data. We then normalize the data using calculated mean and variance across animals.

Neural Pre-Processing

For neural preprocessing, we developed our own, lightweight approach because conventional NMF methods (i) were designed for easy-to-identify, rodent neural cell bodies but do not generalize to tracking our axons, (ii) model action potentials, but do not take into account the distinct dynamical properties of graded potential neurons, and (iii) perform an optimization that, in our hands, is overly sensitive to ini-

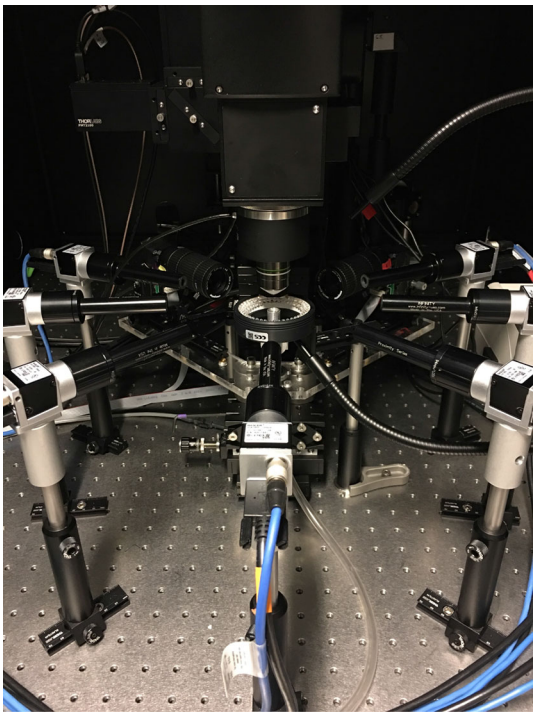

Fig. 8 Recording Setup for the MC2P Dataset. Calibrated infrared cameras form a ring with the animal at its center A two-photon microscope measures the calcium influx while positioned directly above the cameras

tial conditions and computationally very expensive to run on our full dataset. For our approach, data were synchronized using a custom Python package (Aymanns, 2021). We then estimated the motion of the neurons using images acquired on the red (tdTomato) PMT channel. The first image of the first trial was selected as a reference frame to which all other frames were registered. For image registration, we estimated the vector field describing the motion between two frames. To do this, we numerically solved the optimization problem in Eq. 9, where w is the motion field, $\mathcal{I}_t$ is the image being transformed, $\mathcal{I}_r$ is the reference image, and Ω is the set of all pixel coordinates (Chen et al., 2018; Aymanns, 2021).

$$\hat{w} = argmin_w \sum_{x\in\Omega} ||\mathcal{I}_t(x + w(x)) - \mathcal{I}_r(x)||_2^2 - \lambda \sum_{x\in\Omega} ||\nabla w(x)||_2^2 \tag{9}$$

A smoothness promoting parameter λ was empirically set to 800. We then applied $\hat{w}$ to the green PMT channel (GCaMP6s). To denoise the motion corrected green signal, we trained a DeepInterpolation network (Lecoq et al., 2020) for nine epochs for each animal and applied it to the rest of the frames. We only used the first 100 frames of each trial and used the first and last trials as validation data. The batch size was set to 20 and we used 30 frames before and after the current frame as input. In order to have a direct correlation

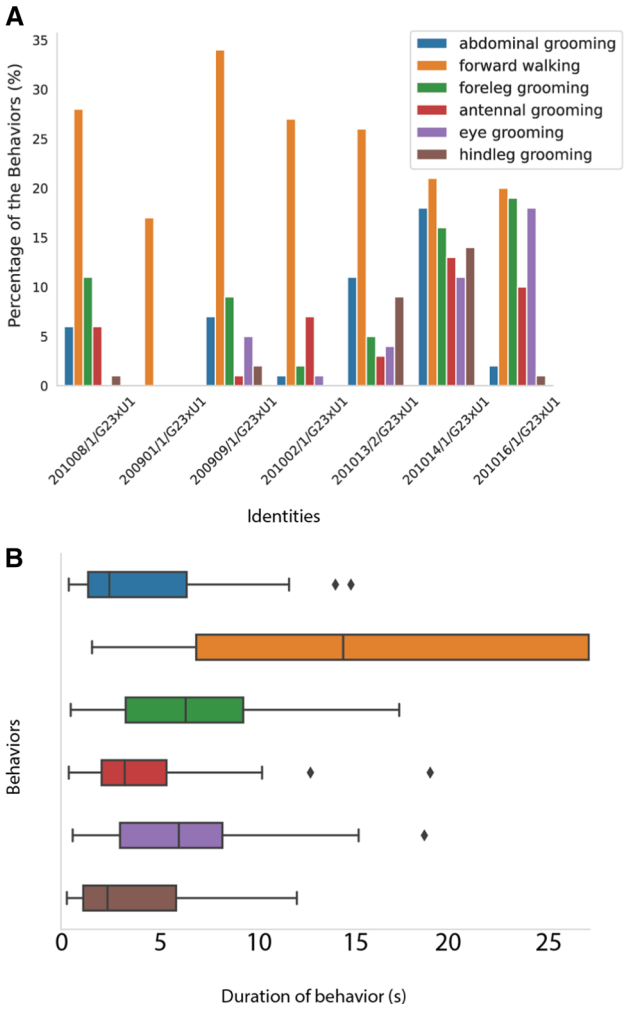

Fig. 9 Motion Capture and two-photon dataset statistics. Visualizing **A** the number of annotations per animal and **B** the distribution of the durations of each behavior across animals. Unlike scripted human behaviors, animal behaviors occur spontaneously. The total number of behaviors and their durations do not follow a uniform distribution, therefore making it harder to model

between pixel intensity and neuronal activity we applied the following transformation to all neural images $\frac{F-F_0}{F_0} \times 100$, where F_0 is the baseline fluorescence in the absence of neural activity. To estimate F_0, we used the pixel-wise minimum of a moving average of 15 frames.

Neural Fluorescence Signal Decay

The formal relationship between the neural image $\mathbf{n}_t$ and neural activity (underlying neural firings) $\mathbf{s}_t$ can be modeled as a first-order autoregressive process

$$\mathbf{n}_t = \gamma \mathbf{n}_{t-1} + \alpha \mathbf{s}_t,$$

where $\mathbf{s}_t$ is a binary variable indicating an event at time t (e.g. the neuron firing an action potential). The amplitudes γ and α determine the rate at which the signal decays and the initial

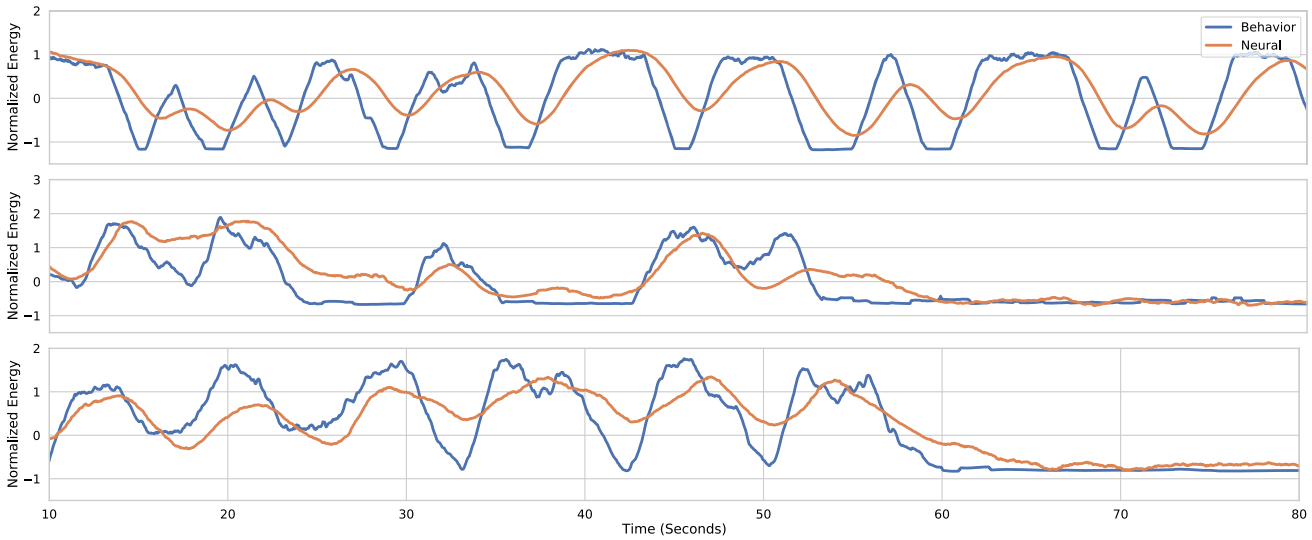

Fig. 10 Visualizing the temporal correlation between behavioral and neural energies on multiple animals. The behavioral and neural energies are calculated as the normalized distances between consecutive frames. The multi-modal energies show a similar temporal pattern. The slower neural energy decay is due to the calcium dynamics

response to an event, respectively. In general, $0 < \gamma < 1$, therefore resulting in an exponential decay of information pertaining to $\mathbf{s}_t$ to be inside of $\mathbf{n}_t$. A single neural image $\mathbf{n}_t$ includes decaying information from previous neural activity, and hence carries information from previous behaviors. For more detailed information on calcium dynamics, see Pnevmatikakis et al. (2013); Rupprecht et al. (2021). Assuming no neural firings, $\mathbf{s}_t = 0$, $\mathbf{n}_t$ is given by $\mathbf{n}_t = \gamma^t \mathbf{n}_0$. Therefore, we define the calcium kernel $\mathcal{K}$ as $\mathcal{K}_t = \gamma^t$.

Dataset Analysis

We show the distribution of annotations across 7 animals and action duration distribution in Appendix Fig. 9. Unlike scripted actions in human datasets, the animal behavior is spontaneous, therefore does not follow a uniform distribution. The average duration of behaviors can also change across behaviors. Walking is the most common behavior and lasts longer than other behaviors. We visualize the correlation between the neural and behavioral energy in Appendix Fig. 10. We quantify the energy as the Euclidean distance between consecutive, vectorized 3D poses. Similarly, for the neural energy, we calculate the Euclidean distance between consecutive images. To be able to compare corresponding energies, we first synchronize neural and behavioral modalities. We then smooth the corresponding time series using Gaussian convolution with kernel size of 11 frames. We observe that there is a strong correlation between the modalities, suggesting large mutual information.

Method Details

Augmentations

Aside from the augmentations mentioned before, for the neural image transformation family $\mathcal{T}_n$, we used a sequential application of Poisson noise and Gaussian blur and color jittering. In contrast with recent work on contrastive visual representation learning, we only applied brightness and contrast adjustments in color jittering because neural images have a single channel that measures calcium indicator fluorescence intensity. We did not apply any cropping augmentation, such as cutout, because action representation is often highly localized and non-redundant (e.g., grooming is associated with the activity of a small set of neurons and thus with only a small number of pixels). We applied the same augmentations to each frame in single sample of neural data.

For the behavior transformation family $\mathcal{T}_b$, we used a sequential application of scaling, shear, and random temporal and spatial dropping. We did not apply rotation and translation augmentations because the animals were tethered (i.e., restrained from moving freely), and their direction and absolute location were fixed throughout the experiment. We did not use time warping because neural and behavioral information are temporally linked (e.g., fast walking has different neural representations than slow walking).

Mixing Parameters

We analyze the effects of mixing individual poses, instead of whole motion sequences, through our mixing augmentation

Table 6 Architecture details

First part of the Neural Encoder f_n				
Layer	# filters	K	S	Output
input	1	–	–	$T \times 128 \times 128$
conv1	19	(3,3,3)	(1,2,2)	$T \times 128 \times 128$
conv2	37	(3,3,3)	(1,2,2)	$T \times 64 \times 64$
conv3	55	(3,3,3)	(1,2,2)	$T \times 32 \times 32$
conv4	73	(3,3,3)	(1,2,2)	$T \times 16 \times 16$
conv5	91	(3,3,3)	(1,2,2)	$T \times 8 \times 8$
conv6	109	(3,3,3)	(1,2,2)	$T \times 4 \times 4$
conv7	128	(3,3,3)	(1,2,2)	$T \times 2 \times 2$
conv8	128	(3,3,3)	(1,1)	$T \times 1 \times 1$
attention9	–	(1,1)	(1,1)	$1 \times 1 \times 128$
fc10	128	(1,1)	(1,1)	$1 \times 1 \times 1$
fc11	128	(1,1)	(1,1)	$T \times 1 \times 1$
Behavioral Encoder f_b				
Layer	# filters	K	S	Output
input	60	–	–	$T \times 60$
conv1	64	(3)	(1)	$T \times 64$
conv2	80	(3)	(1)	$T \times 80$
mp2	–	(2)	(2)	$T/2 \times 80$
conv2	96	(3)	(1)	$T/2 \times 96$
conv2	112	(3)	(1)	$T/2 \times 112$
conv2	128	(3)	(1)	$T/2 \times 128$
attention6	–	(1)	(1)	1×128
fc7	128	(1)	(1)	1×128

Shown are half of the neural encoder f_n and behavior encoder f_b functions. How these encoders are used is shown in Fig. 3. Both encoders produce 128 dimensional output, while first half of the neural encoder do not downsample on the temporal axis. *mp* denotes a max-pooling layer. Batch Normalization and ReLU activation are added after every convolutional layer

in Fig. 7. We compare the distribution similarity across individuals when tested on single poses and windows of poses. We observe that the distribution similarity across individuals in behavioral modality is much larger in pose level when compared to the whole motion sequence, therefore making it easier to mix behavioral data in pose level. We quantify the distribution similarity using MMD (Mean Maximum Discrepancy) and Homogeneity metrics. Similarly, mixing individual poses decreases the overall change in the motion sequence, as quantified by the Mean Squared Distance. Yet, the degree to which identity information is hid does not strongly correlate with the window size of mixing. Therefore, overall, suggesting mixing in pose level is better than mixing whole motion sequences.

Implementation Details

For all methods, we initialized the weights of the networks randomly unless otherwise specified. To keep the experiments consistent, we always paired 32 frames of neural data with 8 frames of behavioral data. For the neural data, we used a larger time window because the timescale during which dynamic changes occur are smaller. For the paired modalities, we considered data synchronized if their center frames had the same timestamp. We trained contrastive methods for 200 epochs and set the temperature value τ to 0.1. We set the output dimension of $\mathbf{z}_b$ and $\mathbf{z}_n$ to 128. We used a cosine training schedule with three epochs of warm-up. For non-contrastive methods, we trained for 200 epochs with a learning rate of $1e{-}4$, and a weight decay of $1e{-}5$, using the Adam optimizer (Kingma & Ba, 2015). We ran all experiments using an Intel Core i9-7900X CPU, 32 GB of DDR4 RAM, and a GeForce GTX 1080. Training for a single SimCLR network for 200 epochs took 12 hours. To create train and test splits, we removed two trials from each animal and used them only for testing. We used the architecture shown in Appendix Table 6 for the neural image and behavioral pose encoder. Each layer except the final fully-connected layer was followed by Batch Normalization and a ReLU activation function (Ioffe & Szegedy, 2015). For the self-attention mechanism in the behavioral encoder (Appendix Table 6), we implement Bahdanau attention (Bahdanau et al., 2015). Given the set of intermediate behavioral representations $S \in \mathbb{R}^{T \times D}$, we first calculated,

$$\mathbf{r} = W_2 \tanh\left(W_1 S^\top\right), \quad \mathbf{a}_i = -\log\left(\frac{\exp\left(\mathbf{r}_i\right)}{\sum_j \exp\left(\mathbf{r}_j\right)}\right)$$

where W_1 and W_2 are a set of matrices of shape $\mathbb{R}^{12 \times D}$ and $\mathbb{R}^{1 \times 12}$ respectively. $\mathbf{a}_i$ is the assigned score i-th pose in the sequence of motion. Then the final representation is given by $\sum_i^T \mathbf{a}_i S_i$. For the projection function g_n and g_b we use a simple 2 layer MLP.

Supplementary Videos

Motion Capture and Two-Photon (MC2P) Dataset: The following videos are sample behavioral-neural recordings from two different flies. The videos show (left) raw behavioral RGB video together with (right) registered and denoised neural images at their original resolutions. The behavioral video is resampled and synchronized with the neural data. The colorbar indicates normalized relative intensity values. Calculation of $\Delta F/F$ is previously explained under Dataset Collection section.

Video 1: https://drive.google.com/file/d/1-xjiwn7qgiou3_nf0nlyu7549KKlSfUx

Video 2: https://drive.google.com/file/d/1DUOzSbbE8uPdPNvXChJWYb9bN9AyUfuO

Action Label Annotations: Sample behavioral recordings from multiple animals using a single camera. Shown are eight

different action labels: *forward walking*, *pushing*, *hindleg grooming*, *abdominal grooming*, *foreleg grooming*, *antennal grooming*, *eye grooming* and *resting*. Videos are temporally down-sampled. Animals and labels are randomly sampled.
Video 3: https://drive.google.com/file/d/1cnwRRyDZ4crrVVxRBbx32Za-vlxSP7sy
Animal Motion Capture: Sample behavioral recordings with 2D poses from six different camera views. Each color denotes a different limb. The videos are temporally down-sampled for easier view.
Video 4: https://drive.google.com/file/d/1uYcL7_Zl-N0mlG1VTrg67s2Cy71wml5S
Video 5: https://drive.google.com/file/d/1eMcP-Ec1c4yBQpC4CNv45py7gObmuUeA

References

Abbaspourazad, H., Choudhury, M., Wong, Y. T., Pesaran, B., & Shanechi, M. M. (2021). Multiscale low-dimensional motor cortical state dynamics predict naturalistic reach-and-grasp behavior. *Nature Communications, 12*(1), 607.

Aymanns, F. (2021). ofco: optical flow motion correction. https://doi.org/10.5281/zenodo.5518800.

Aymanns, F. (2021). utils2p. https://doi.org/10.5281/zenodo.5501119.

Aymanns, F., Chen, C-L., & Ramdya, P. (2022). Descending neuron population dynamics during odor-evoked and spontaneous limb-dependent behaviors. *Neuroscience*. https://doi.org/10.7554/eLife.81527.

Bahdanau, D., Hyun C. K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Bala, P. C., Eisenreich, B. R., Yoo, S. B. M., Hayden, B. Y., Park, H. S., & Zimmermann, J. (2020). Automated markerless pose estimation in freely moving macaques with openmonkeystudio. *Nature Communications, 11*(1), 4560.

Batty, E., Whiteway, M., Saxena, S., Biderman, D., Abe, T., Musall, S., et al. (2019). Behavenet: Nonlinear embedding and bayesian neural decoding of behavioral videos. *Advances in Neural Information Processing Systems (NeurIPS),* .

Berthelot, D., Carlini, N., Goodfellow, I., Oliver, A., & Papernot, N., Raffel, C. (2019). MixMatch: A holistic approach to semi-supervised learning.

Bohnslav, J. P., Wimalasena, N. K., Clausing, K. J., Dai, Y. Y., Yarmolinsky, D. A., Cruz, T., et al. (2021). Deepethogram, a machine learning pipeline for supervised behavior classification from raw pixels. *eLife, 10,* e63377.

Bolaños, L. A., Xiao, D., Ford, N. L., LeDue, J. M., Gupta, P. K., Doebeli, C., et al. (2021). A three-dimensional virtual mouse generates synthetic training data for behavioral analysis. *Nature Methods, 18*(4), 378–381.

Cande, J., Namiki, S., Qiu, J., Korff, W., Card, G. M., Shaevitz, J. W., et al. (2018). Optogenetic dissection of descending behavioral control in Drosophila. *eLife, 7,* 970.

Chen, M.-H., Kira, Z., AlRegib G., Yoo J., Chen, R., & Zheng, J. (2019). Temporal attentive alignment for large-scale video domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Chen, K., Gabriel, P., Alasfour, A., Gong, C., Doyle, W. K., Devinsky, O., et al. (2018). Patient-specific pose estimation in clinical environments. *IEEE Journal of Translational Engineering in Health and Medicine, 6,* 1–11.

Chen, C. L., Hermans, L., Viswanathan, M. C., Fortun, D., Aymanns, F., Unser, M., et al. (2018). Imaging neural activity in the ventral nerve cord of behaving adult drosophila. *Nature Communications, 9,* 1–10.

Collinger, J. L., Gaunt, R. A., & Schwartz, A. B. (2018). Progress towards restoring upper limb movement and sensation through intracortical brain-computer interfaces. *Current Opinion in Biomedical Engineering, 8,* 84–92.

Dave, I., Gupta, R., Rizve, M. N., & Shah, M. (2021). TCLR: Temporal contrastive learning for video representation. *arXiv*.

Dombeck, D. A., Khabbaz, A. N., Collman, F., Adelman, T. L., & Tank, D. W. (2007). Imaging large-scale neural activity with cellular resolution in awake, mobile mice. *Neuron, 56*(1), 43–57.

Ecker, A. S., Berens, P., Keliris, G. A., Bethge, M., Logothetis, N. K., & Tolias, A. S. (2010). Decorrelated neuronal firing in cortical microcircuits. *Science, 327*(5965), 584–587.

Eyjolfsdottir, E. A. (2017). *Computational Methods for Behavior Analysis*. PhD thesis.

Eyjolfsdottir, E., Branson, S., Burgos-Artizzu, X. P., Hoopfer, E. D., Schor, J., Anderson, D. J., & Perona, P. (2014). Detecting social actions of fruit flies. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Eyjolfsdottir, E., Branson, K., Yue, Y., & Perona, P. (2017). Learning recurrent representations for hierarchical behavior modeling. In *International Conference on Learning Representations, (ICLR)*.

Farshchian, A., Gallego, J., Cohen, J., Bengio, Y., Miller, L., & Solla, S. (2018). Adversarial domain adaptation for stable brain-machine interfaces. *arXiv*.

Ganin, Y., & Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Ganzer, P. D., Colachis, S. C., Schwemmer, M. A., Friedenberg, D. A., Dunlap, C. F., Swiftney, C. E., et al. (2020). Restoring the sense of touch using a sensorimotor demultiplexing neural interface. *Cell, 181*(4), 763-773.e12.

Gao, Y., Archer, E., Paninski, L., & Cunningham, J. (2016). Linear dynamical neural population models through nonlinear embeddings. *Advances in Neural Information Processing Systems (NeurIPS)*.

Giovannucci, A., Friedrich, J., Gunn, P., Kalfon, J., Brown, B. L., Koay, S. A., et al. (2019). Caiman: An open source tool for scalable calcium imaging data analysis. *eLife, 8,* e38173.

Glaser, J. I., Benjamin, A. S., Chowdhury, R. H., Perich, M. G., Miller, L. E., & Kording, K. P. (2020). Machine learning for neural decoding. *eNeuro*, 7(4).

Gosztolai, A., Günel, S., Lobato-Ríos, V., Pietro Abrate, M., Morales, D., Rhodin, H., et al. (2021). Liftpose3d, a deep learning-based approach for transforming two-dimensional to three-dimensional poses in laboratory animals. *Nature Methods, 18*(8), 975–981.

Graving, J. M., Chae, D., Naik, H., Li, L., Koger, B., Costelloe, B. R., & Couzin, I. D. (2019). Deepposekit, a software toolkit for fast and robust animal pose estimation using deep learning. *eLife, 8,* e47994.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A. J. (2006). A kernel method for the two-sample-problem. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*.

Günel, S., Rhodin, H., Morales, D., Campagnolo, J., Ramdya, P., & Fua, P. (2019). Deepfly3D, a deep learning-based approach for 3D

limb and appendage tracking in tethered, adult drosophila. *eLife, 8,* e48571.

He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Ionescu, C., Papava, I., Olaru, V., & Sminchisescu, C. (2014). Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 36,* 1325–1339.

Johnson, R. E., Linderman, S., Panier, T., Wee, C. L., Song, E., Herrera, K. J., et al. (2020). Probabilistic models of larval zebrafish behavior reveal structure on many scales. *Current Biology, 30*(1), 70-82e4.

Kang, G., Jiang, L., Wei, Y., Yang, Y., & Hauptmann, A. G. (2020). Contrastive adaptation network for single-and multi-source domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Karashchuk, P., Rupp, K. L., Dickinson, E. S., Walling-Bell, S., Sanders, E., Azim, E., et al. (2021). Anipose: A toolkit for robust markerless 3d pose estimation. *Cell, 36*(13), 109730.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations, (ICLR)*.

Koger, B., Deshpande, A., Kerby, J. T., Graving, J. M., Costelloe, B. R., & Couzin, I. D. (2022). Multi-animal behavioral tracking and environmental reconstruction using drones and computer vision in the wild. *bioRxiv*.

Kostas, D., Aroca-Ouellette, S., & Rudzicz, F. (2021). BENDR: Using transformers and a contrastive self-supervised learning task to learn from massive amounts of EEG data. *Frontiers in Human Neuroscience, 15,* 253.

Kulkarni, N., Gupta, A., Fouhey, D. F, & Tulsiani, S. (2020). Articulation-aware canonical surface mapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 452–461.

Lacourse, K., Yetton, B., Mednick, S., & Warby, S. C. (2020). Massive online data annotation, crowdsourcing to generate high quality sleep spindle annotations from eeg data. *Scientific Data, 7*(1), 190.

Lau, C. K. S., Jelen, M., & Gordon, M. D. (2021). A closed-loop optogenetic screen for neurons controlling feeding in drosophila. *G3 (Bethesda), 11*(5), 05.

Lecoq, J., Oliver, M., Siegle, J. H., Orlova, N., & Koch, C. (2020). Removing independent noise in systems neuroscience data using deepinterpolation. *bioRxiv*.

Lee, K., Zhu, Y., Sohn, K., Li, C.-L., Shin, J., & Lee, H. (2021). i-mix: A domain-agnostic strategy for contrastive representation learning. In *International Conference on Learning Representations, (ICLR)*.

Li, S., Günel, S., Ostrek, M., Ramdya, P., Fua, P., & Rhodin, H. (2020). Deformation-aware unpaired image translation for pose estimation on laboratory animals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, W., Ji, S., Chen, X., Kuai, B., He, J., Peng, Z., & Li, Q. (2020). Multi-source domain adaptation for decoder calibration of intracortical brain-machine interface. *Journal of Neural Engineering, 17,* 10.

Lin, L., Song, S., Yang, W., & Liu, J. (2020). MS2L: Multi-task self-supervised learning for skeleton based action recognition. In *Proceedings of the ACM International Conference on Multimedia*.

Linderman, S., Johnson, M., Miller, A., Adams, R., Blei, D., & Paninski, L. (2017). Bayesian learning and inference in recurrent switching linear dynamical systems. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Linderman, S., Nichols, A., Blei, D., Zimmer, M., & Paninski, L. (2019). Hierarchical recurrent state space models reveal discrete and continuous dynamics of neural activity in c. elegans. *bioRxiv*.

Liu, Y., & Yan, Q., Alahi, A. (2020). Social nce: Contrastive learning of socially-aware motion representations. arXiv.

Lobato-Rios, V., Gizem Özdil, P., Ramalingasetty, S. T., Arreguit, J., Ijspeert, A. J., & Ramdya, P. (2021). Neuromechfly, a neuromechanical model of adult drosophila melanogaster. *bioRxiv*.

Mao, X., Ma, Y., Yang, Z., Chen, Y., & Li, Q. (2019). Virtual mixup training for unsupervised domain adaptation. *arXiv*.

Mohsenvand, M. N., Izadi, M. R., & Maes, P. (2020). Contrastive representation learning for electroencephalogram classification. In *Proceedings of the Machine Learning for Health NeurIPS Workshop*.

Munro, J., & Damen, D. (2020). Multi-modal domain adaptation for fine-grained action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Musall, S., Kaufman, M. T., Juavinett, A. L., Gluf, S., & Churchland, A. K. (2019). Single-trial neural dynamics are dominated by richly varied movements. *Nature Neuroscience, 22*(10), 1677–1686.

Nakagome, S., Luu, T. P., He, Y., Ravindran, A. S., & Contreras-Vidal, J. L. (2020). An empirical comparison of neural networks and machine learning algorithms for eeg gait decoding. *Nature Scientific Reports, 10*(1), 4372.

Nassar, J., Linderman, S. W., Bugallo, M., & Park, I.-S. (2019). Tree-structured recurrent switching linear dynamical systems for multi-scale modeling. *arXiv*.

Nath, T., Mathis, A., Chen, A. C., Patel, A., Bethge, M., & Mathis, M. W. (2019). Using deeplabcut for 3d markerless pose estimation across species and behaviors. *Nature Protocols, 14*(7), 2152–2176.

Overman, K., Choi, D., Leung, K., Shaevitz, J., & Berman, G. (2021). Measuring the repertoire of age-related behavioral changes in drosophila melanogaster. *bioRxiv*.

Palazzo, S., Kavasidis, I., Kastaniotis, D., Dimitriadis, S. I. (2018). Recent advances at the brain-driven computer vision workshop 2018. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*.

Palazzo, S., Spampinato, C., Kavasidis, I., Giordano, D., Schmidt, J., & Shah, M. (2021). Decoding brain representations by multimodal learning of neural activity and visual features. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 43,* 3833–3849.

Pan, T., Song, Y., Yang, T., Jiang, W., & Liu, W. (2021). Videomoco: Contrastive video representation learning with temporally adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Pandarinath, C., O'Shea, D. J., Collins, J., Jozefowicz, R., Stavisky, S. D., Kao, J. C., et al. (2018). Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature Methods, 15,* 805–815.

Pedersen, M., Haurum, J. B., Bengtson, S. H., & Moeslund, T. B. (2020). 3d-zef: A 3d zebrafish tracking benchmark dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Pei, F., Ye, J., Zoltowski, D., Wu, A., Chowdhury, R.H., Sohn, H., O'Doherty, J.E., Shenoy, K.V., Kaufman, M.T., Churchland, M., Jazayeri, M., Miller, L. E., Pillow, J., Park, I. M., Dyer, E. L., & Pandarinath, C. (2021). Neural latents benchmark '21: Evaluating latent variable models of neural population activity.

Pereira, T. D., Tabris, N., Li, J., Ravindranath, S., Papadoyannis, E. S., Wang, Z. Y., Turner, D. M., McKenzie-Smith, G., Kocher, S. D., Falkner, A. L., Shaevitz, J. W., & Murthy, M. (2020). Sleap: Multi-animal pose tracking. *bioRxiv*.

Pereira, T. D., Shaevitz, J. W., & Murthy, M. (2020). Quantifying behavior to understand the brain. *Nature Neuroscience, 23*(12), 1537–1549.

Peterson, S. (2021). Ecog and arm position during movement and rest.
Peterson, S. M., Rao, R. P. N., & Brunton, B. W. (2021). Learning neural decoders without labels using multiple data streams. *bioRxiv*.
Pnevmatikakis, E. A., Merel, J., Pakman, A., & Paninski, L. (2013). Bayesian spike inference from calcium imaging data. *arXiv*.
Robie, A. A., Hirokawa, J., Edwards, A. W., Umayam, L. A., Lee, A., Phillips, M. L., et al. (2017). Mapping the neural substrates of behavior. *Cell, 170*(2), 393-406.e28.
Rupprecht, P., Carta, S., Hoffmann, A., Echizen, M., Blot, A., Kwan, A. C., et al. (2021). A database and deep learning toolbox for noise-optimized, generalized spike inference from calcium imaging. *Nature Neuroscience, 24*(9), 1324–1337.
Sahoo, A., Panda, R., Feris, R. S., Saenko, K., & Das, A. (2020). Select, label, and mix: Learning discriminative invariant feature representations for partial domain adaptation. *arXiv*.
Sahoo, A., Shah, R., Panda, R., Saenko, K., & Abir, D. (2021). Contrast and mix: Temporal contrastive video domain adaptation with background mixing. arXiv.
Sanakoyeu, A., Khalidov, V., McCarthy, M. S., Vedaldi, A., & Neverova, N. (2020). Transferring dense pose to proximal animal classes.
Sani, O. G., Abbaspourazad, H., Wong, Y. T., Pesaran, B., & Shanechi, M. M. (2021). Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification. *Nature Neuroscience, 24*(1), 140–149.
Sani, O. G., Yang, Y., Lee, M. B., Dawes, H. E., Chang, E. F., & Shanechi, M. M. (2018). Mood variations decoded from multi-site intracranial human brain activity. *Nature Biotechnology, 36*(10), 954–961.
Seelig, J. D., Chiappe, M. E., Lott, G. K., Dutta, A., Osborne, J. E., Reiser, M. B., & Jayaraman, V. (2010). Two-photon calcium imaging from head-fixed Drosophila during optomotor walking behavior. *Nature Methods, 7*(7), 535–540.
Segalin, C., Williams, J., Karigo, T., Hui, M., Zelikowsky, M., Sun, J. J., Perona, P., Anderson, D. J., & Kennedy, A. (2020). The mouse action recognition system (mars): A software pipeline for automated analysis of social behaviors in mice. *bioRxiv*.
Segalin, C., Williams, J., Karigo, T., Hui, M., Zelikowsky, M., Sun, J. J., et al. (2021). The mouse action recognition system (mars) software pipeline for automated analysis of social behaviors in mice. *Elife, 10,* e63720.
Shen, Z., Liu, Z., Liu, Z., Savvides, M., Darrell, T., & Xing, E. (2022). Un-mix: Rethinking image mixtures for unsupervised visual representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.
Shenoy, K. V., & Kao, J. C. (2021). Measurement, manipulation and modeling of brain-wide neural population dynamics. *Nature Communications, 12*(1), 633.
Singh, S. H., Peterson, S. M., Rao, R. P. N., & Brunton, B. W. (2021). Mining naturalistic human behaviors in long-term video and neural recordings. *Journal of Neuroscience Methods, 358,* 109199.
Spampinato, C. et al. (2017). Deep learning human mind for automated visual classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
Stringer, C., Pachitariu, M., Steinmetz, N., Reddy, C. B., Carandini, M., & Harris, K. D. (2019). Spontaneous behaviors drive multidimensional, brainwide activity. *Science, 364*(6437), 255–255.
Su, K., Liu, X., & Shlizerman, E. (2020). Predict & cluster: Unsupervised skeleton based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
Sun, J. J., Karigo, T., Chakraborty, D., Mohanty, S. P., Wild, B., Sun, Q., Chen, C., Anderson, D. J., Perona, P., Yue, Y., & Kennedy, A. (2021). The multi-agent behavior dataset: Mouse dyadic social interactions. In *Advances in Neural Information Processing Systems (NeurIPS)*.
Sun, J. J, Kennedy, A., Zhan, E., Anderson, D. J, Yue, Y., & Perona, P. (2021). Task programming: Learning data efficient behavior representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
Topalovic, U., Aghajan, Z. M., Villaroman, D., Hiller, S., Christov-Moore, L., Wishard, T. J., et al. (2020). Wireless programmable recording and stimulation of deep brain activity in freely moving humans. *Neuron, 108*(2), 322-334.e9.
Urai, A. E., Doiron, B., Leifer, A. M., & Churchland, A. K. (2021). Large-scale neural recordings call for new insights to link brain and behavior. *arXiv*.
Utsumi, K., Takano, K., Okahara, Y., Komori, T., Onodera, O., & Kansaku, K. (2018). Operation of a p300-based brain-computer interface in patients with duchenne muscular dystrophy. *Scientific Reports, 8*(1), 1753.
van den Oord, A., Li, Y., & Vinyals, O. (2019). Representation learning with contrastive predictive coding. *arXiv*.
Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., & Bengio, Y. (2019). Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Learning Representations, (ICLR)*.
Walter, T., & Couzin, I. D. (2021). Trex, a fast multi-animal tracking system with markerless identification, and 2d estimation of posture and visual fields. *eLife, 10,* e64000.
Wang, X., Farhadi, A., Rao, R. P. N., & Brunton, B. W. (2018). Ajile movement prediction: Multimodal deep learning for natural human neural recordings and video. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.
Wei, J., & Zou, K. (2019). EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.
Wen, S., Yin, A., Tseng, P.-H., Itti, L., Lebedev, M. A., & Nicolelis, M. (2021). Capturing spike train temporal pattern with wavelet average coefficient for brain machine interface. *Scientific Reports, 11*(1), 19020.
Wilbur, A. H., Ronnie, B., Bharadwaj, H. M., & Siskind, J. M. (2021). Object classification from randomized eeg trials. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
Wiltschko, A. B., Johnson, M. J., Iurilli, G., Peterson, R. E., Katon, J. M., Pashkovski, S. L., et al. (2015). Mapping sub-second structure in mouse behavior. *Neuron, 88,* 1121–1135.
Wu, A., Buchanan, E. K., Whiteway, M., Schartner, M., Meijer, G., Noel, J.-P., Rodriguez, E., Everett, C., Norovich, A., Schaffer, E., Mishra, N., Salzman, C. D., Angelaki, D., & Bendesky, A. (2020). The International Brain Laboratory The International Brain Laboratory, John P Cunningham, and Liam Paninski. Deep graph pose: a semi-supervised deep graphical model for improved animal pose tracking. In *Advances in Neural Information Processing Systems (NeurIPS)*.
Xu, Y., Yang, J., Cao, H., Mao, K., Yin, J., & See, S. (2021). Aligning correlation information for domain adaptation in action recognition. *arXiv*.
Yuan, X., Lin, Z., Kuen, J., Zhang, J., Wang, Y., Maire, M., Kale, A., & Faieta, B. (2021). Multimodal contrastive training for visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., & Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision (ICCV)*.

Zhang, H., Cissé, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations, (ICLR)*.

Zhang, Y., Jiang, H., Miura, Y., Manning, C. D., & Langlotz, C. P. (2020). Contrastive learning of medical visual representations from paired images and text. *arXiv*.